Neural Comput & Applic (2015) 26:263–276
DOI 10.1007/s00521-014-1632-y

ADVANCES IN INTELLIGENT DATA PROCESSING AND ANALYSIS

Probabilistic ensemble Fuzzy ARTMAP optimization using hierarchical parallel genetic algorithms

Chu Kiong Loo · Wei Shiung Liew · Manjeevan Seera · Einly Lim

Received: 30 November 2013 / Accepted: 17 May 2014 / Published online: 13 June 2014

Abstract In this study, a comprehensive methodology for overcoming the design problem of the Fuzzy ARTMAP neural network is proposed. The issues addressed are the sequence of training data for supervised learning and optimum parameter tuning for parameters such as baseline vigilance. A genetic algorithm search heuristic was chosen to solve this multi-objective optimization problem. To further augment the ARTMAP's pattern classification ability, multiple ARTMAPs were optimized via genetic algorithm and assembled into a classifier ensemble. An optimal ensemble was realized by the inter-classifier diversity of its constituents. This was achieved by mitigating convergence in the genetic algorithms by employing a hierarchical parallel architecture. The best-performing classifiers were then combined in an ensemble, using probabilistic voting for decision combination. This study also integrated the disparate methods to operate within a single framework, which is the proposed novel method for creating an optimum classifier ensemble configuration with minimum user intervention. The methodology was benchmarked using popular data sets from UCI machine learning repository.

C. K. Loo (✉) · W. S. Liew · M. Seera
Faculty of Computer Science and Information Technology,
University Malaya, Kuala Lumpur, Malaysia
e-mail: ckloo.um@gmail.com; ckloo.um@um.edu.my

W. S. Liew
e-mail: liew.wei.shiung@gmail.com

M. Seera
e-mail: mseera@um.edu.my

E. Lim
Faculty of Biomedical Engineering, University Malaya,
Kuala Lumpur, Malaysia
e-mail: einly_lim@um.edu.my

Keywords Ensembles · Fuzzy ARTMAP · Plurality voting · Parallel genetic algorithm · Feature selection

1 Introduction

Fuzzy ARTMAP (FAM) neural networks [1] are commonly used for pattern classification problems. Under a supervised learning condition, the FAM is able to form complex correlations between a given multi-dimensional input or exemplars and a multi-dimensional output or category labels. However, the classification accuracy is dependent on the parameter settings and the order in which exemplars were presented during supervised learning. Finding a combination of parameters that will yield the best classification performance is essentially an optimization task.

FAM have been used in a wide range of applications. In fault diagnosis of rolling element bearings, a FAM ensemble based on the improved Bayesian belief method is used [2]. A Gaussian ARTMAP model is proposed by Mokhtar et al. [3] for building better and more efficient energy management systems. Simplified FAM model is used in detecting faults of induction motor, with the inputs of transient current signals [4]. A FAM neural network is employed by Liang et al. [5] in classifying and grading yarn surface qualities with satisfactory performance. For fault detection and diagnosis in a power generation plant, a FAM network with evolutionary programming is proposed, with consistent experimental results [6].

The FAM optimization task is required to explore several different characteristics for optimum classifier performance. The parameter settings of the FAM control the fundamental architecture of the classifier and affect its ability to learn patterns during the training process.

In addition, the internal knowledge base of the FAM grows and changes shape with each additional exemplar presented to the neural network during training. A biased training sequence, or one which involves poorly defined exemplars, may affect the FAM's classification ability. The neural network is somewhat able to compensate for the sequencing problem if the exemplars are presented repeatedly during training [7], but the problem will be accumulative for FAM classifiers relying on online learning. A number of methods were proposed for overcoming the shortfalls of the FAM, such as parameter tuning [8–11], ensemble learning [12, 13], or by modifying the neural network architecture to create variants of the FAM [7, 14–17].

Using the FAM neural network, the proposed framework consisted of the following methods; given a classification task with a data set of pattern examples, parameter tuning was performed, using genetic algorithms (GA) to search for effective combinations of parameters and training sequence for creating a trained FAM classifier. The GA is able to converge onto optimal points in the search space through competitive eliminations and refining the search in between the surviving candidates. With the goal of developing an ensemble of classifiers, the role of the GA in this framework was to generate and iteratively evolve a population of FAMs not only for improved classification accuracy, but for diversity between individual classifiers. Traditional single-population GA have a tendency to converge on a single optimum point, mainly due to the elitist selection process. Convergence in this case is undesirable due to the intention to create a classifier ensemble.

Ensemble learning has recently gained much attention in various learning tasks such as classification, clustering, and regression problems [18]. A hybrid model consisting of Fuzzy Min–Max (FMM) neural network and the random forest (RF) model, comprising an ensemble of classification and regression trees (CART) for condition monitoring of induction motors is developed by Seera et al. [19]. Using a similar model, FMM, CART, and RF ensemble is used in three benchmark medical data sets, with positive outcome from the experiments [20]. An artificial neural network ensemble, composed using a multi-class classifier is used for traffic sign recognition and hand-written digit recognition, with good results [21].

The rationale for an ensemble is to combine complementary information from multiple diverse classifiers to achieve a classification accuracy higher than any individual classifier. A variant GA known as hierarchical fair-competition parallel genetic algorithms (HFCPGA) was proposed in [22], where genetic convergence was mitigated by distributing chromosomes across multiple subpopulations instead of a single population. The populations are independent of each other and are arranged in a hierarchy so that candidates will compete only against other candidates with similar levels of fitness.

Diversity was considered an important factor when designing ensembles of classifiers. Zenobi and Cunningham [23] discussed the possibility of having an ensemble of suboptimum classifiers outperforming an ensemble of optimum classifiers due to successful trade-off between diversity and accuracy. To ensure the GA generates a diverse set of FAM configurations, a number of methods were implemented in the framework such as the HFCPGA model and feature subset selection. Feature subset selection, wherein the FAM is trained using only a partial representation of the training patterns, would reduce the complexity of the resultant trained FAM, otherwise known as the problem of "overfitting" [24]. Thus, instead of having an ensemble of overfitted classifiers, the ensemble would consist of classifiers that were each trained on a small subset of the training data set. Several studies showed the advantages of an ensemble of diverse and specialized FAMs trained using feature-selected data sets [25–28]. For this framework, feature subsets generated by the GA would be used to filter the training data set before being used for training the FAM.

Having generated a diverse population of FAMs, an ensemble was created using classifiers with the best classification accuracy. The final ensemble will be integrated by means of a probabilistic voting strategy [29, 30] that assigns each classifier with a reliability index and skewing ensemble decisions in favor of high-reliability classifiers while reducing the voting weights of classifiers with poor accuracy. When classifying an object, the reliability of the classification was computed from the probabilistic output from each classifier. This adds a new dimension to classification output from the ensemble, in which reliability represents the "confidence" of the ensemble in its prediction.

Similar research has been conducted in the various literatures. Radtke et al. [31] proposed a similar two-step method for optimizing classifier training using feature subsets, and for crafting ensembles of classifiers, both using different evolutionary algorithm methodologies. Using an automated method for selecting classifiers for an ensemble may be considered for future development of our existing framework. Ishibuchi et al.'s methodology for generating ensembles of fuzzy rule-based classifiers [32] utilized a multi-objective GA for finding Pareto-optimum combinations of accuracy, and size and complexity of the fuzzy rule sets, whereas the method proposed in this study selected the FAMs which showcased the highest classification accuracy regardless of the internal configuration.

In this work, the FAM classifier requires training sequence and parameter optimization to perform with peak accuracy. A GA was suggested for this task, using a hierarchical and parallelized architecture to reduce homogeneity and to distribute convergence over multiple points.

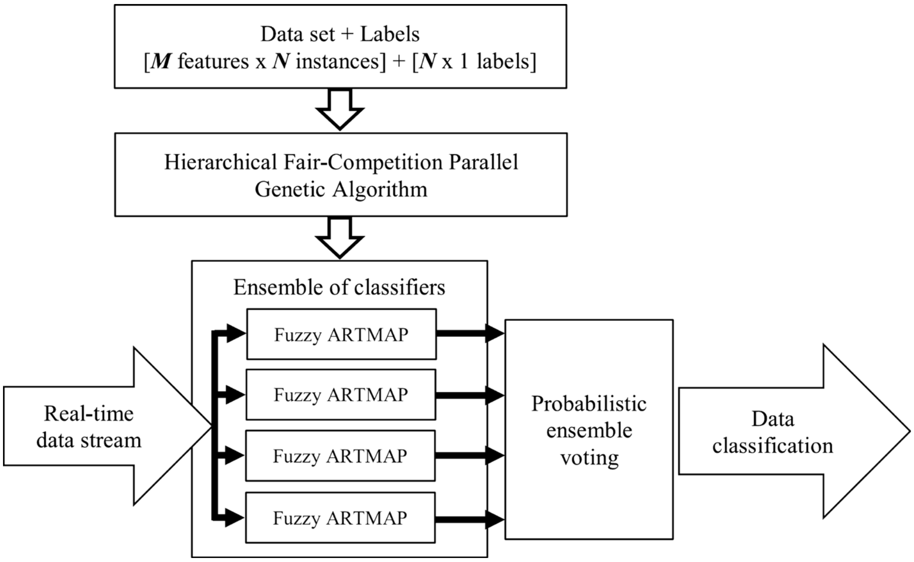
Data set + Labels
[M features x N instances] + [N x 1 labels]
Hierarchical Fair-Competition Parallel Genetic Algorithm
Ensemble of classifiers
Fuzzy ARTMAP
Fuzzy ARTMAP
Fuzzy ARTMAP
Fuzzy ARTMAP
Real-time data stream
Probabilistic ensemble voting
Data classification

Fig. 1 Chart of the proposed method for creating an ensemble of optimized Fuzzy ARTMAP classifiers

A population of optimal FAMs was generated, from which an ensemble of classifiers was created. Ensemble decision was based on probabilistic voting of the individual classifiers, weighted according to each FAMs reliability. The framework was proposed as an automated method for generating an ensemble of trained FAM classifiers for a given pattern classification task with consistently high accuracy. The workings of the FAM classifier, the genetic optimization methods, and the probabilistic voting scheme is detailed in Sect. 2. Section 3 details the experimental setup, where several data sets were used for benchmark. Results and discussions are included in Sect. 4. Concluding remarks is given in Sect. 5.

2 Description of framework

The proposed framework for generating an optimized ensemble of classifiers consist of three parts: classifier optimization, ensemble creation, and classifier combination, as illustrated in Fig. 1. Given a pattern classification task with a data set of labeled exemplars, the algorithm first generates a population of random candidates, or chromosomes, each of which can be used to create a single trained FAM classifier. The hierarchical parallel genetic optimization method searches for optimum chromosomes through genetic selection, crossover, and mutation, iterated across several generations from the initial starting population. Subsequently, the new population of candidates will consist of optimized chromosomes from which the best were selected to create an ensemble of FAM classifiers. Pattern classification was performed by the ensemble by having the constituent FAMs classify an unknown object. Individual classification decisions were combined using probabilistic voting to select the final ensemble decision.

The individual modules of the framework consist of the FAM pattern classifier, a HFCPGA for classifier optimization and a probabilistic voting step to combine classifier decision in an ensemble.

2.1 Fuzzy ARTMAP

Fuzzy ARTMAP neural networks operate on the principle of adaptive resonance theory for creating correlating connections between a given multi-dimensional input vector and the corresponding output representing its class category. The FAM neural network architecture is illustrated in Fig. 2. The system consists of two ART modules with a mapping field connecting both modules. Given a set of exemplars and their respective category labels, the input vectors are fed into the ART$_a$ input module while the labels go to the ART$_b$ output module. The supervised learning method modifies the weighted connections between F_2^a, F^{ab}, and F_2^b to achieve resonance between the input and output. Repeated over multiple times with a variety of exemplars, the FAM learns to recognize and classify similar input objects into the most likely class label based on its learning experiences.

Details of the FAM operations [1] is given in further details, as follows:

1. Given an object to be classified, in the form of a normalized vector a with M attributes in the range of 0 to 1.
2. Vector a along with its complement, $a^C = 1 - a$, was encoded as a single input object A:

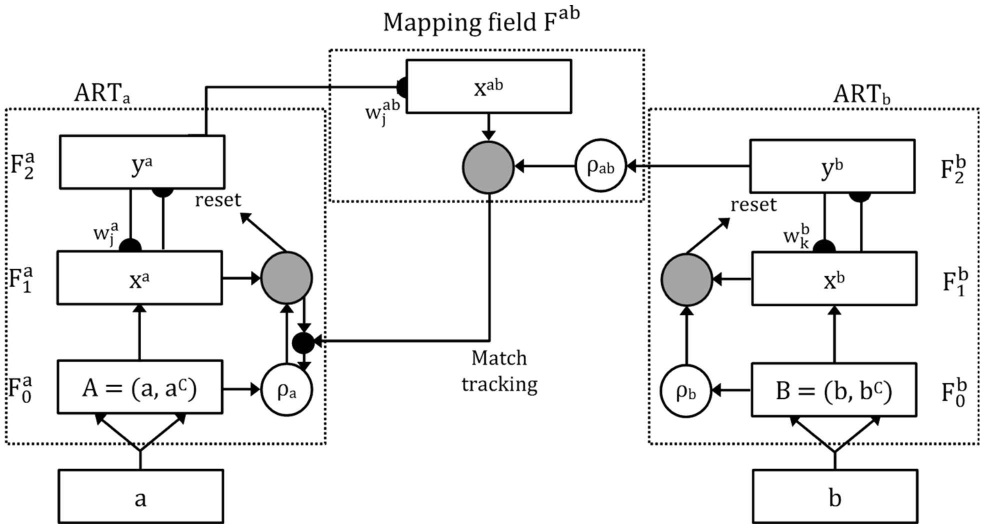

Fig. 2 Structure of the Fuzzy ARTMAP neural network

$$A = (a, a^C) \tag{1}$$

3. Among the nodes in F_2^a that have not been selected, a node J was selected with the maximum choice function:

$$T_j = |A \wedge w_j| + (1 - \alpha)(M - |w_j|) \tag{2}$$

Uncommitted nodes were initialized with all values of w_J set to 1.

4. The selected node was matched against the bottom-up input A. The field F_1^a represented the fuzzy intersection between the input vector A and the weights of the node, w_J. The vector representing the match between input vector A and the selected node weights w_J is represented as:

$$x = A \wedge w_J \tag{3}$$

where $\wedge$ denotes the component-wise minimum, or fuzzy intersection, of the bottom-up input vector A and the top-down expectation w_J. At this point, one of several cases may occur:

- Node J failed to meet the match criterion: $\frac{|x|}{|A|} < \rho_a$. Another node was chosen and Step 3 was repeated.
- Node J meets the match criterion: $\frac{|x|}{|A|} \geq \rho_a$. The node was used to make a classification prediction for object A.
  - The object A was transmitted along the weighted connections between F_2^a and the mapping field F^{ab}. A successful map between the input A and output B was determined by the map field match criterion:

$$|x^{ab}| \geq \rho_{ab}|y_b| \tag{4}$$

  - The match tracking equation was designed to trigger a mismatch reset if the selected node J makes a wrong prediction:

$$\frac{d\rho_a}{dt} = -(\rho - \bar{\rho}) + \Gamma R r^c \tag{5}$$

In the case of an incorrect prediction, the predictive error parameter R is set to 1 and the current vigilance parameter ρ_a was incremented according to Eq. 5 until ρ_a was larger than the match value $\frac{|x|}{|A|}$, thus failing the match criterion. The algorithm then loops back to select a new node J and repeat Step 3. In the meantime, ρ_a decays by the match tracking parameter, ϵ before the next node J was selected. This mechanism was designed to minimize predictive errors by stimulating search between nodes, while maximizing the network's generalization ability through manipulating the current vigilance parameter.

  - In the case where object a was successfully mapped to class b, or if the selected node J was uncommitted, the system learns by incorporating the input object A into node J:

$$w_J^{\text{new}} = (1 - \beta)w_J^{\text{old}} + \beta(w_J^{\text{old}} \wedge A) \tag{6}$$

- The algorithm loops back to Step 1 for the next object to be classified.

The pseudocode of the process is given below.

```
TRAIN($DataSet, \alpha, \beta, \epsilon, \bar{\rho}$)
for $i \leftarrow 1$ to $N$ objects in DataSet
    do
       $A = [a_i, a_i^C]$
       $\rho_a = \bar{\rho}$
       for $j \leftarrow 1$ to $J$
           do
              while $(T_j = |A \wedge w_j| + (1-\alpha)(M - |w_j|) \geq T_{max})$ and $(\frac{|A \wedge w_j|}{|A|} \leq \rho_a)$
                  do
                     if $J$ is uncommitted
                        then $w_J = \{w_1^J, w_2^J, ..., w_{2M}^J\} = \{1, 1, ..., 1\}$
                     if ($J$ is uncommitted) or ($|x^{ab}| \geq \rho_{ab}|y_b|$)
                        then and $w_J^{new} = (1-\beta)w_J^{old} + \beta(w_J^{old} \wedge A)$
                        else
                             while $|x^{ab}| < \rho_{ab}|y_b|$
                                 do
                                    $\frac{d\rho_a}{dt} = -(\rho - \bar{\rho}) + \Gamma R r^c$
              $\rho_a = \rho_a - \epsilon$
```

The ART-based neural network was dependent on its internal configuration of node weights, which in turn were affected by a number of factors such as the ARTMAP parameter settings and the training data used for learning. A number of approaches for ARTMAP optimization used evolutionary algorithms to search for the optimum training sequence [11] and ARTMAP parameter settings [33]. Kaylani et al. [16], however, proposed using multi-objective GA for optimizing directly the ARTMAP's internal topology, thus bypassing the need to determine the parameters.

In this study, however, we will focus on using GA for optimizing the training order and the ARTMAP parameters, given as:

- Baseline vigilance, $\bar{\rho}$. Setting zero vigilance allows a greater degree of generalization, while setting high vigilance only permits learning from highly specific exemplars.
- Choice parameter, α. Influences the degree of uniqueness of each committed node.
- Learning rate, β. Determines how quickly the nodes adapt and learn the given data.
- Match tracking parameter, ϵ. Determines the rate in which current vigilance returns to baseline after each predictive error by the selected node.

2.2 Hierarchical fair-competition parallel genetic algorithm

Genetic algorithms (GAs) are search heuristics employing the principles of natural evolution to search for and refine solutions for a given optimization task. A candidate solution to the problem is encoded in the form of a string of genes, each representing one variable, constituting into a single chromosome. A population of chromosomes are maintained over the course of several consecutive generations, during which the chromosomes are subject to competitive eliminations followed by genetic reproduction to create new chromosomes variants from the remaining survivors.

GAs have been used in the previous literature for optimizing classifier performance, such as generic fuzzy rule-based classifiers [32], radial basis function networks [34], and simple vector regression [35]. In all cases, optimizing a classifier requires the GA to search for an optimum combination of parameters, with each classifier having different parameters than the others.

One problem usually encountered especially with GAs with high turnover rate, is that of population convergence. With each successive round of eliminations and reproduction, the number of chromosomes sharing common genetic traits will increase, leading to the population clustering around a relatively small area in the solution space. Therein exists the possibility of premature convergence around a local optimum. Furthermore, in the context of this study, single-point convergence is counterproductive as it results in the final group of chromosomes to be homogeneous. For classifier ensembles, inter-classifier diversity is considered an important trait in order for the ensemble to perform better than its constituent classifiers. The default GA methodology is therefore unsuitable for our purpose to optimize individual FAMs for an ensemble of classifiers.

The HFCPGA [22] was conceived as a technique for mitigating the issue of premature convergence. The HFCPGA employs multiple populations of chromosomes, each evolving independently of each other. Distribution of chromosomes in each population is arranged in a hierarchy, ensuring that the chromosomes in each population face fair competition against chromosomes with similar fitness. The multitude of

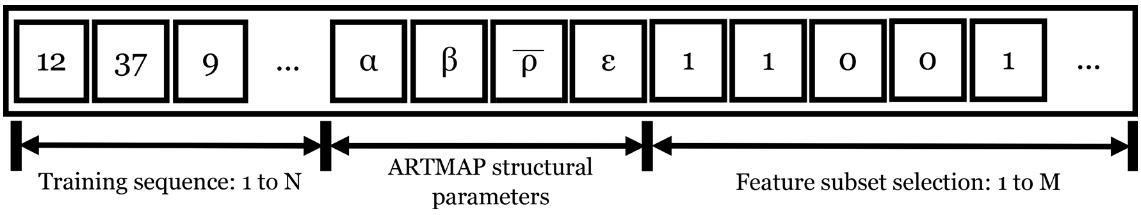
12
37
9
...
α
β
ρ̄
ε
1
1
0
0
1
...
Training sequence: 1 to N
ARTMAP structural parameters
Feature subset selection: 1 to M

Fig. 3 A single chromosome consisting of three sections comprising all the parameters for optimizing FAMs for the given pattern classification task

communities improves global diversity by splitting the genetic convergence across multiple populations. Periodic immigration lets high-fitness chromosomes to move to a higher hierarchy and vice-versa, thus injecting new diversity into the converging populations.

A single chromosome contains all of the variables to be optimized, encoded as a single numerical string as shown in Fig. 3. In this case, a chromosome consists of three parts required to generate an initial FAM architecture and train it to recognize and classify patterns.

- Training sequence. Given a data set with N exemplars, this string is a random sequence of numbers from 1 to N representing the order in which the exemplars will be presented to the FAM during supervised learning.
- ARTMAP structural parameters α, β, ϵ, and $\bar{\rho}$, randomized according to the parameter range defined in Table 1.
- Feature subset selection. Given a data set of exemplars with M attributes, this M-length binary string decides which attribute to be excluded from the training data set. A user-defined parameter can be implemented to ensure a minimum and/or maximum limit to the number of attributes in the subset.

Figure 4 shows the flowchart of the genetic optimization process, elaborated as follows:

1. A population of random chromosomes was generated, each representing a single configuration for creating and training a FAM.
2. Each chromosome was used to generate a single trained FAM classifier.
3. Each FAM was fitness-tested using tenfold cross-validation. The training data set was divided evenly into ten folds. For a single iteration, onefold was set aside while the remaining folds were used for training the classifier. The trained classifier was then used for classifying the remaining fold. Repeat ten times, each time using a different fold for testing.
4. The fitness of the tested chromosome was computed from the mean recognition rate of the resulting trained FAM classifier.

Table 1 Variables and parameters in the FAM architecture

Variable	Description
a	Input object to be classified. Formatted as a numerical vector ranging from 0 to 1
a^C	Complement-coded vector of a. Each element in the vector is $a_i^C = 1 - a_i$
α	Choice parameter. Determines node selection for signal function T_j. Range from [0, 1], default $\alpha = 0.01$
A	Combination of original input vector a and its complement a^C
b	The true class category in which object a belongs to
β	Learning rate, or rate in which node weights were updated. Range from [0, 1], default $\beta = 1.0$ for fast learning
Γ	Fraction additive in vigilance parameter
ϵ	Match tracking parameter. Rate in which vigilance decays to baseline. Range from $[-1, +1]$, default ϵ =-0.001
J	Selected critical feature pattern. Encoded as a vector of weights w_{ij}
r	Mismatch parameter. $r = 1$ if $\rho\|A\| - \|x\| > 0$. Default $r = 0$
R	Predictive error parameter. $R = 1$ if node J makes a predictive error. Default $R = 0$
ρ	Current vigilance parameter. Range from [0, 1]
$\bar{\rho}$	Baseline vigilance parameter. Range from [0, 1]
T_j	Choice-by-difference signal function. Nodes in F_2^a are selected in order of highest to lowest signal function
x^a	Intersection between J and A: $x^a = \|A \wedge w_J\|$

5. In the first generation, chromosomes were grouped evenly into subpopulations according to similar fitness. For subsequent generations, a selected chromosome was immigrated to an adjacent subpopulation if it possessed a significantly higher or lower fitness than the average of the subpopulation.
6. Genetic selection, reproduction, and mutation were performed.

   - A number of chromosomes with the least fitness were discarded.
   - Offspring chromosomes were generated to replace discarded chromosomes. Reproduction was performed to generate an offspring which inherited the

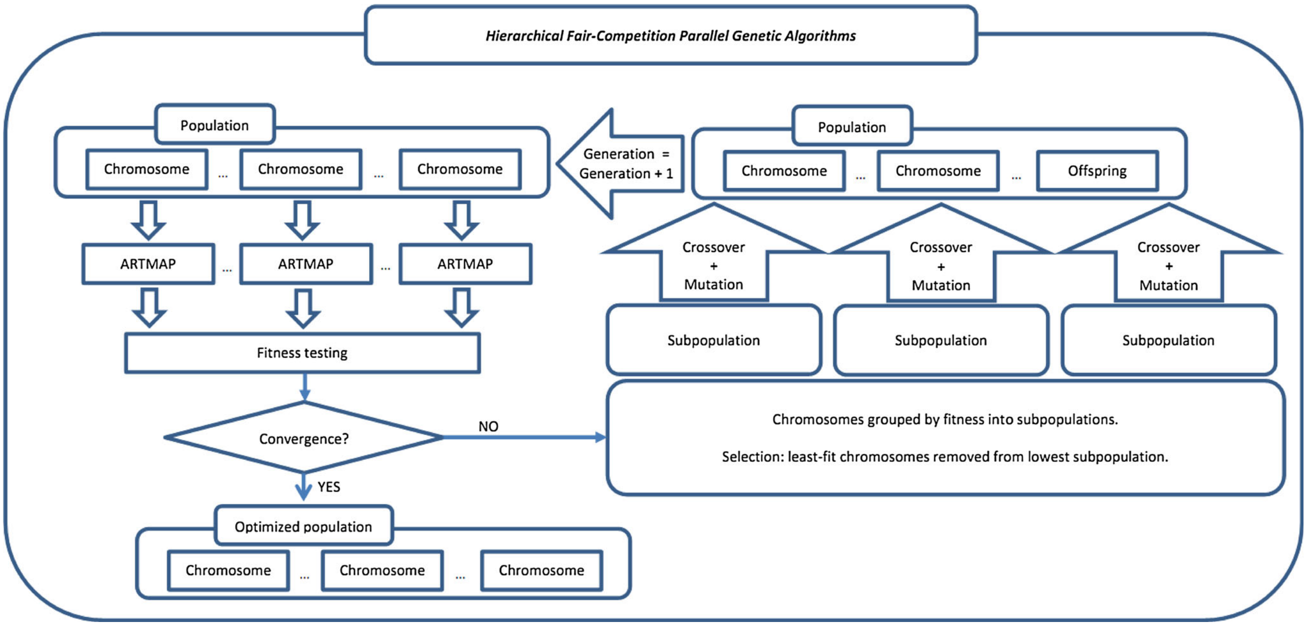

Fig. 4 Flowchart of the hierarchical parallel genetic algorithms

traits of parent chromosomes, followed by a mutation to add a random element to any selected gene in the chromosome. The reproduction and mutation operations, outlined in Fig. 5, were performed slightly differently for each segment of the chromosome.

- The newly generated offspring chromosome was added into the lowest subpopulation until they were given a chance to migrate after the next round of fitness testing.

7. The generation counter was incremented, and steps 2–5 were repeated until convergence was achieved by reaching a maximum generation limit.

The following Table 2 summarizes the key differences between the methodology of the HFCPGA and a simple GA.

2.3 Probabilistic voting

Classifier ensembles operate on the assumption that each member of the ensemble functions as an independent expert, and that combining the complementary information from every classifier, the ensemble is able to score a higher classification accuracy than any of its constituent classifiers. The decision combination method used in this study is similar the probabilistic voting system developed by Loo and Rao [30] based on the work by Lin et al. [29].

Given an ensemble with L classifiers $\{E_1, E_2, \ldots, E_L\}$ for classifying an input object X into one of k class categories $\{C_1, C_2, \ldots, C_k\}$.

$$P(E_i(X) = C(X)) = p_i \tag{7}$$

$E_i(X)$ is the class prediction of classifier E_i for the given object X, and $C(X)$ is the object's true class. Thus, each classifier has a constant recognition rate p_i to classify an object correctly. In the case of an incorrect recognition, it was assumed that all residual classes have equal probability of being selected:

$$P(E_i(X) = C_j) = (1 - p_i)/(k - 1) = e_i \tag{8}$$

where $j = 1, 2, \ldots, k$ and $C_j \neq C(X)$. Also, assuming classifier independence:

$$P(E_i(X), E_2(X), \ldots, E_L(X) | C(X) = C_j) = \prod_{i=1}^{L} P(E_i(X) | C(X) = C_j) \tag{9}$$

To minimize the error rate of the combination system, the class C_j with the largest *a posteriori* probability should be selected according to the Bayes' rule:

$$P(C(X) = C_j | E_1(X), E_2(X), \ldots, E_L(X)) = \frac{\left[\prod_{i=1}^{L} P(E_i(X) | C(X) = C_j)\right] \times P(C(X) = C_j)}{P(E_1(X), E_2(X), \ldots, E_L(X))} \tag{10}$$

$$P(E_i(X) | C(X) = C_j) = \begin{cases} p_i & \text{if } E_i(X) = C_j \\ e_i & \text{if } E_i(X) \neq C_j \end{cases} = e_i \left(\frac{p_i}{e_i}\right)^{\delta_{ij}(X)} \tag{11}$$

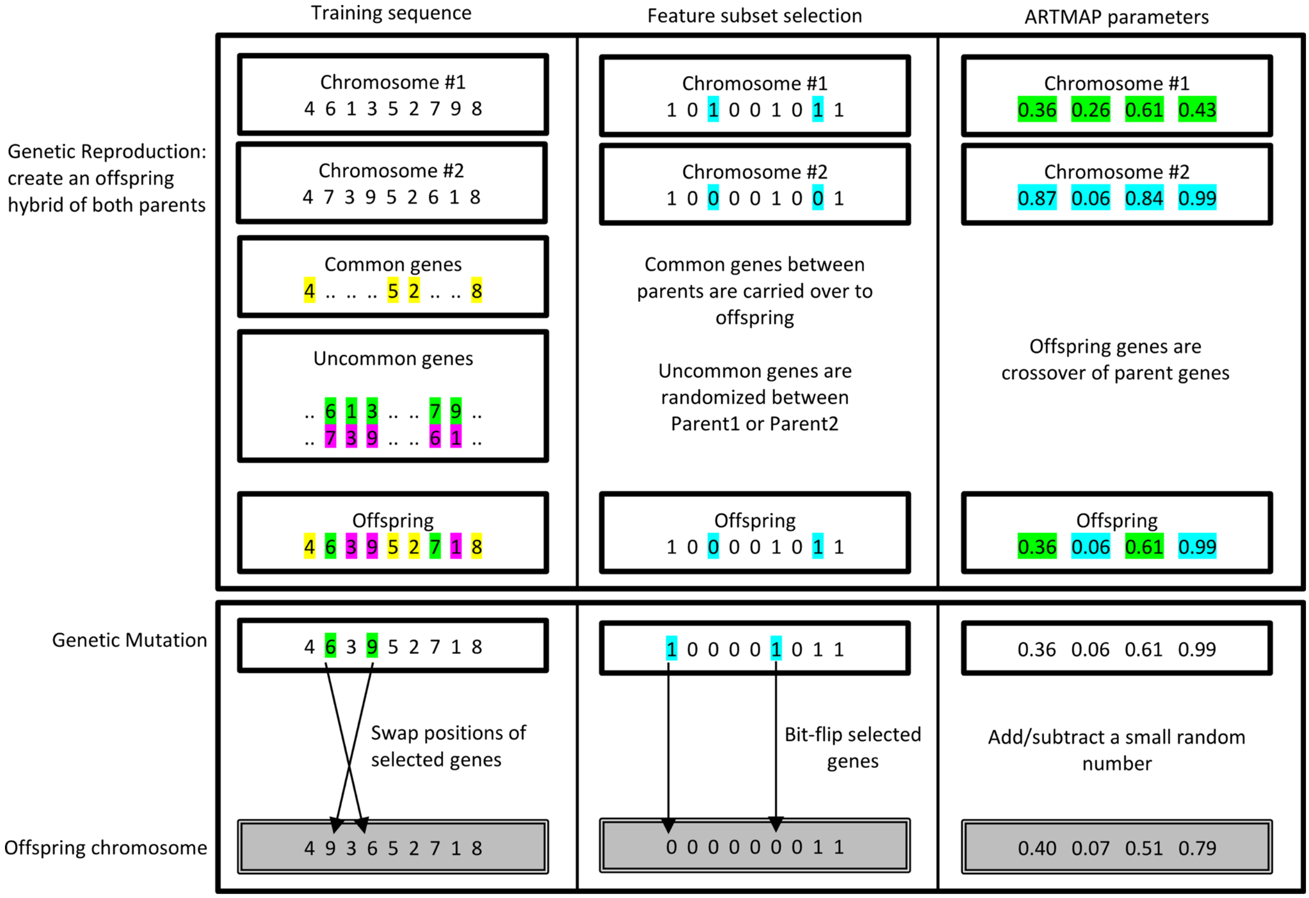

Fig. 5 Examples of genetic reproduction and mutation for the different segments of the chromosome

Table 2 Methodological differences between simple GA and HFCPGA

Simple GA	HFCPGA
Single population of chromosomes	Chromosomes divided into multiple subpopulations
All chromosomes placed in the same population	Chromosomes were grouped into subpopulations according to fitness
For reproduction, two parent	Roulette-wheel selection chooses a subpopulation
chromosomes were selected at random	Two parent chromosomes were chosen at random from the same subpopulation
No migration	Two chromosomes from different subpopulations swap positions if a high-fitness chromosome was located in a low-fitness subpopulation and vice-versa

where

$$\delta_{ij}(x) = \begin{cases} 1 & \text{if } E_i(X) = C_j \\ 0 & \text{if } E_i(X) \neq C_j \end{cases}$$

$$\begin{aligned} P(C(X) = C_j | E_1(X), E_2(X), \ldots, E_L(X)) \\ = \frac{\left[\prod_{i=1}^{L} e_i \left(\frac{p_i}{e_i}\right)^{\delta_{ij}(X)}\right] \times P(C(X) = C_j)}{P(E_1(X), E_2(X), \ldots, E_L(X))} \end{aligned} \tag{12}$$

Let

$$\begin{aligned} Y(X) &= \frac{\left(\prod_{i=1}^{L} e_i\right)}{P(E_1(X), E_2(X), \ldots, E_L(X))}, \\ P(C(X) &= C_j | E_1(X), E_2(X), \ldots, E_L(X)) \\ &= Y(X)\left[P(C(X) = C_j) \prod_{i=1}^{L} \left(\frac{p_i}{e_i}\right)^{\delta_{ij}(X)}\right] \end{aligned} \tag{13}$$

As $Y(X)$ is the same for every class category, the effective decision function is the second part:

$$\begin{aligned} D_j(X) &= \ln P(C(X) = C_j) + \sum_{i=1}^{L} \ln\left(\frac{p_i}{e_i}\right) \delta_{ij}(X) \\ &= \ln P(C(X) = C_j) + \sum_{i=1}^{L} \ln\left(\frac{(k-1)p_i}{(1-p_i)}\right) \delta_{ij}(X) \end{aligned} \tag{14}$$

Equation 14 is the generic form of the probabilistic voting rule. Each classifier can have a different weight, and each class has a constant *a priori* probability. Probabilistic voting as in Eq. 14 is equivalent to the Bayesian criterion under the conditions:

- Classifier decisions are independent of each other.
- Misclassifications are evenly distributed among the $k - 1$ residual classes.
- In the case of a tie, one of the classes with the maximum support is selected.

Practically, the assumption of classifier independence is difficult to achieve. Most classifiers will be prone to simultaneous misclassifications for difficult exemplars. Loo and Rao [30] proposed a modified methodology consisting of two independent situations: Anumber of classifiers have a probability α of simultaneously misclassifying a given object. Otherwise, the classifiers have a probability $(1 - \alpha)$ to perform independently. Using this model, the overall recognition rate of classifier E_i is $(1 - \alpha)p_i$. The α parameter can be tweaked depending on the reliability of the given classifiers' predictive accuracy.

The use of probabilistic votes as opposed to winner-take-all allows a measure of reliability for class selection [30]. The reliability term used here can be defined as the classifier's confidence in its prediction and is measured as the difference between the probabilistic score of the best class and the score of the second-best class. Thus, in the case when a majority of FAMs misclassify an input with a small margin of reliability, the ensemble may still be able to select the correct classification if the remaining FAMs have higher reliability in their predictions.

3 Experimental setup

An experiment was designed to test the performance of the classifier ensembles created using the methods outlined in previous sections. The experiment was proposed as follows: given a predetermined training data set, the GA was employed to find the optimum combinations of the following: Parameters used for initializing the FAM architecture; a single-pass training sequence, in which each instance is only presented to the FAM once during supervised learning; and a subset of features of the data set that will be presented to the FAM. The data sets used for testing were obtainable from the UCI machine learning repository [36]. Details of the data sets are given in Table 3. Data sets were selected for their variety of instance sizes, feature set sizes, as well as binary and multi-classed classification problems. All of the data sets have been used in the past literature on various classification methods, thus allowing comparison in terms of classification accuracy.

Table 3 Data sets used for benchmarking

Data set	Instances	Attributes	Classes
Australian	640	14	2
Glass	214	9	6
Heart	270	13	2
Hepatitis	155	19	2
Ionosphere	351	34	2
Iris	150	4	3
Satellite	6,435	36	7

The user-defined parameters are given in Table 4. Population size is the number of chromosomes that are maintained at any one time. This value should be set large enough to maintain genetic variety and to avoid local optima convergence, but not so large that the algorithm would be slowed down by fitness testing. A larger population size would reduce the number of generations for the GA to reach the global optima. With five subpopulations maintained for the HFCPGA, each subpopulation would contain ten chromosomes each. For genetic operations, mutation rate serves to introduce an element of randomness. Setting a high mutation rate (i.e., close to 1) means that the GA is essentially performing a random search, whereas a very low mutation rate inhibits exploration, limited to only the direct offspring of the current batch of chromosomes. The generation limit determines how many iterations of selection, reproduction, and mutation to perform before stopping. The limit should be large enough for the system to be able to approach the global optimum.

The GA implements an elitism selection scheme, where a number of chromosomes with the lowest fitness was discarded and replaced by the offspring of the remaining chromosomes. A higher rejection rate implies greater selective pressure, as essentially a large percentage of the population is replaced at each generation. For the HFCPGA, this typically means that the lower subpopulations would have a high turnover rate. Setting a low rejection rate would slow down exploration, whereas setting a high rejection rate would limit genetic variety of the population to only the offspring of a small group of chromosomes. The feature fraction parameter limits each FAM to learn from a subset of the training data's feature space. Setting a low feature limit allows more combinations of specializations in specific feature subsets. Having a diverse selection of FAMs with narrow specializations would benefit the classifier ensemble, as posited in a number of literature [25–28].

Table 4 Parameters used for the genetic optimization phase

Parameters	Description	Default
Subpopulations	Number of sub-populations to maintain	5
Chromosomes	Total number of chromosomes	50
Generations	Maximum number of generations	50
Feature fraction	Maximum fraction of selected features	0.5
Mutation rate	Fraction of genes to modify	0.1
Rejection	Fraction of population to discard per generation	0.5

The settings for this experiment were selected arbitrarily

Assuming a pattern classification task with a training data set of M attributes × N instances:

1. A population of chromosomes were generated. Each chromosome consists of FAM parameters with random settings, a random subset of M attributes, and a random sequence of N instances.
2. Fitness testing for each chromosome involved using the parameters defined in the chromosome to create and train a single FAM classifier:
   (a) An initial FAM classifier was generated with the ARTMAP parameters defined in the chromosome.
   (b) A copy of the training data set was created and rearranged to match the training sequence and to remove any attribute not selected in the feature subset.
   (c) A tenfold cross-validation scheme was proposed for the experiment. The instances in the data set were divided into ten groups, each containing an equal number of instances. Ten FAMs were first initialized using the ARTMAP parameters. Then for each FAM, nine groups of instances will be used for training while the tenth group was reserved for testing. For each of the ten FAMs, a different testing group was used, while supervised learning was performed using the instances from the nine groups following the sequence determined by the chromosome. This method was intended to ensure that no instance will be given to the FAM twice, and that the FAM's accuracy would be validated by instances that was not used during training.
   (d) Fitness of the tested chromosome was calculated as the percentage of correctly classified instances, averaged from the tenfold cross-validation tests.
3. Given the initial population, two parallel tracks were performed. One population was optimized using HFCPGA, while another identical population was optimized using simple GA.
4. In the case of HFCPGA, the population of chromosomes were sorted by fitness into subpopulations. A number of chromosomes with the lowest fitness were discarded and replaced with offspring generated from survivors.
   (a) A roulette-wheel method was used for selecting a subpopulation, giving a higher chance of selection to low-fitness subpopulations to explore less fit solutions. From the selected subpopulation, two chromosomes were selected at random.
   (b) Genetic crossover, reproduction and mutation was performed to create a single offspring chromosome with shared genetic traits from both parent chromosomes.
   (c) Steps 3a and 3b repeated until all discarded chromosomes have been replaced.

   Otherwise in the case of simple GA, random chromosome pairs were chosen to generate new offspring.
5. Steps 2 to 4 iterated up to 50 consecutive generations. The final population of chromosomes were considered as candidates for best classifier parameter settings for this particular pattern classification task.
6. A classifier ensemble was created by selecting the chromosomes with the best fitness. The output classifications of the ensemble was then used for calculating ensemble accuracy.

For fitness testing, tenfold cross-validation was used to ensure that the trained FAM classifier was able to generalize against independent and unknown data, i.e., data that were not used for training the FAM.

4 Results and discussion

4.1 Benchmark data sets

Two optimization methods were performed in parallel, using the same starting population of chromosomes. After completing classifier optimization, the chromosomes from the population were selected to form an ensemble, starting with chromosomes with the highest fitness. Odd numbered sizes of ensembles were created to minimize voting ties. The ensemble was then tested on classification accuracy. The experiment was performed ten times for each data set, and each ensemble size for both GA and HFCPGA optimization methods. The results of the tested classification accuracy was then bootstrapped 5,000 times at 95 % confidence interval, and the mean was recorded in Table 5.

Table 5 Comparison of classification accuracy for other ensemble methods tested against the proposed framework

	Australian	Glass	Heart	Hepatitis	Ionosphere	Iris
Dzeroski [37]	0.8623	0.7537	0.8470	0.8503	0.9345	0.9580
Partalas [38]	-	-	-	0.8614	0.9203	0.9533
Zhang [39]	-	0.8747	0.7987	0.8419	0.9254	0.8780
Ordonez [40]	0.8667	0.7290	-	0.8450	0.9342	-
Wang [41]	0.8607	0.7167	0.8348	-	0.9500	0.9640
Initial Pop.	0.6964	0.6008	0.6909	0.7397	0.8871	0.8320
GA: best1	0.8602	0.6977	0.8017	0.8556	0.9504	0.9667
GA: best3	0.8616	0.6916	0.8266	0.8740	0.9600	0.9705
GA: best5	0.8637	0.6842	0.8364	0.8727	0.9590	0.9695
GA: best7	0.8625	0.6819	0.8401	0.8775	0.9587	0.9667
GA: best9	0.8633	0.6831	0.8422	0.8733	0.9621	0.9667
HFCPGA: best1	0.8607	0.6827	0.7989	0.8513	0.9457	0.9668
HFCPGA: best3	0.8603	0.6807	0.8243	0.8759	0.9610	0.9681
HFCPGA: best5	0.8663	0.7145	0.8387	0.8876	0.9668	0.9616
HFCPGA: best7	0.8651	0.7324	0.8466	0.8905	0.9704	0.9600
HFCPGA: best9	0.8661	0.7347	0.8481	0.8920	0.9690	0.9625

Two different genetic algorithm techniques were performed to observe the effect of the hierarchical parallel architecture. Multiple ensembles were created to observe the effect of ensemble size on the overall classification accuracy. The experiment was performed ten times, and the results were used for generating a bootstrap of 5,000 samples to calculate the bootstrapped mean

The table presents the bootstrapped mean classification accuracy for ensembles of FAM classifiers, comparing the classifier optimization methods using GA and using HFCPGA. As a benchmark, results from other classifier ensemble methods in the literature were also reported, all using tenfold cross-validation. Initial Pop. represents the bootstrapped mean fitness of the initial population of randomly generated chromosomes.

Significance test was performed using Wilcoxon signed rank test [42] to determine whether two groups of data were significantly different. If the p value is lower than a threshold (0.05, or a 95 % confidence interval), the null hypothesis is rejected and the outcome is considered significant. In this case, the null hypothesis is that the performance of GA and HFCPGA ensembles are similar. The hypothesis was tested using the ensemble classification accuracy for each ensemble size and for each data set, calculated from the experiment which was repeated ten times. The obtained p values are shown in Table 6.

In 50 % of the compared examples, HFCPGA ensembles showed better performance, whereas 33 % of the examples showed better performance from GA ensembles. In particular, HFCPGA ensembles performed better than GA with larger ensemble sizes. This may indicate that while the GA was quicker to locate the global optimum solution, the HFCPGA-generated solutions are more widely distributed across the solution space, resulting in better ensemble performance.

Using genetic algorithm for FAM optimization is a time-consuming exercise involving fitness testing every single newly generated chromosome at every generation. Processing time increased exponentially when involving data sets with increasingly large number of instances. Table 7 shows the total time taken for FAM optimization for each data set and the average time taken for each generation. The optimization algorithm was performed in MATLAB using the Parallel Processing Toolbox to distribute the workload across eight processor cores.

4.2 Satellite data set

The framework was tested using a large data set taken from the UCI machine learning repository. The Landsat Satellite data set consisted of 4,435 multi-classed training data samples and 2000 testing data samples. Classifier optimization was performed using the training data samples and HFCPGA, generating a population of high-accuracy FAMs. The FAMs with the best fitness were then selected to form a classifier ensemble for classifying the test data samples.

Classifier optimization using the training data set took approximately 6 h from start to finish, with an average of seven minutes per generation. Ensembles was then constructed using the FAMs with the highest accuracy. After constructing ensembles of various sizes, testing was performed using the testing data set. The experiment was repeated ten times, and the results were used to bootstrap an array of 5,000 samples with 95 % confidence interval. The bootstrapped mean and standard deviation for each ensemble size are given in Table 8.

Table 6 *p* values for paired sign tests for GA vs. HFCPGA ensembles

Ensemble sizes	Australian	Glass	Heart	Hepatitis	Ionosphere	Iris
1	0.0020	0.0020	0.0039	0.0020	0.0020	0.4453
3	0.1055	0.0020	0.0020	0.0996	0.2402	0.0020
5	0.2754	0.0020	0.0059	0.0020	0.0020	0.0020
7	0.0371	0.0020	0.0020	0.0020	0.0020	0.0020
9	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020

Table 7 Total time for FAM optimization for 50 generations and average time for each generation

Data set	Average time per generation	Total time spent (s)
Australian	24.78	1,239
Glass	3.03	151
Heart	4.57	229
Hepatitis	1.69	84
Ionosphere	7.07	354
Iris	1.48	74

Table 8 Ensemble classification accuracy for satellite data set

Ensemble size	Bootstrap mean	Bootstrap SD
Initial Pop.	0.6996	0.0218
1	0.8092	0.0043
3	0.8584	0.0023
5	0.8719	0.0020
7	0.8798	0.0016
9	0.8834	0.0005

4.3 Discussions

The experiments purpose was to benchmark this methodology against other methods for pattern classification. The main thrust of the methodology was to augment the base FAM pattern classifier by pre-classification optimization and combining decisions from multiple diverse FAMs. Optimizing training sequence and FAM parameters as well as random feature subset selection for each FAM improved classification accuracy by a significant percentage compared to the average classification accuracy of the initial population. The initial population of chromosomes can be likened to a group of FAMs created using randomized training data, FAM parameters, and feature subsets, representing the performance of a basic unoptimized FAM classifier.

The ensembles generated using the current member selection scheme does not always produce an improvement over individual classifiers. This may be due to the classifiers having a higher number of correlated misclassifications than correct classifications. As demonstrated for the Iris data set, when individual FAMs each achieved very good classification accuracy, the effect of the naive ensemble selection compounded the errors resulting in poor ensemble performance. Additional correlated FAMs added to the ensemble will only serve to distort the ensemble's classification accuracy. As posited earlier by Zenobi and Cunningham [23], naively selecting the top-performing FAMs in the population for an ensemble was counterproductive, resulting in worse accuracy than the best individual. The results of the study revealed the need for an effective classifier selection for ensemble creation. A review of literature revealed various methods for ensemble selection including using GA [43, 44] or particle swarm optimization [45], or using selective rules for constructing ensembles such as negative correlation [46]. An ensemble selection strategy may be considered for implementation in future versions of the framework.

Feature subset selection operates on the assumption that for a given training data set with M number of attributes, there exists only a small subset of attributes which are highly correlated to the classification of each training instance. The rest of the uncorrelated attributes, under supervised learning, provide noisy information that will serve to negatively affect the classifier's ability to correctly label the given training data. Deriving the most correlated attributes is not as simple as calculating directly the correlation indices. Each attribute in the training data set may not be independent, but interacting with one or more attributes that may not be clearly visible. Evaluation of a feature subset selection was performed indirectly by observing the classification accuracy of the FAM classifier. Feature subset selection in this experiment is optimized by a genetic algorithm, based on Yang et al.'s strategy [24], using binary weights to select and deselect attributes. The feature subset parameter in the experiment indicates the maximum number of attributes that can be selected at any one time, with a minimum of one. Thus, setting the parameter to 0.5 ensured that no classifier will be trained with more than 50 % of the data set's attributes.

A limitation of the proposed method was the significant processing time, the majority of which were for evaluating the fitness function of each chromosome in the classifier optimization step. More detailed examination revealed that a majority of the processing time was spent for training the

FAM classifier. Processing time increased exponentially when data sets with a large number of instances were used. Attempts to reduce total processing time were implemented by reducing the need for repeated validation of the same chromosomes across multiple generations, and by structuring the algorithm to run in parallel using MATLAB's Parallel Processing Toolbox. Using eight parallel threads, the system was able to perform up to four times faster.

The main motivation of using HFCPGA over simple GA was to improve the quality of the population of FAMs for ensemble selection. The results of this experiment were promising but inconclusive in proving the superiority of HFCPGA in that regard. This may be partly caused by the naive voter selection method which selects only the best-fitness FAMs from the population, effectively limiting the selection to the classifiers from the top sub-population. The ensembles may yield improved results if the FAM classifiers were selected using other criteria.

5 Conclusions

The proposed framework integrated several methodologies to create a unified methodology for creating optimum classifier ensembles. A genetic algorithm was employed for efficient searching for the best configuration of parameters such as training sequence, ARTMAP vigilance and learning rate, and feature subset. The best configurations were then used for creating a group of FAM classifiers integrated into an ensemble of classifiers.

Two genetic algorithm methods were used to generate populations of optimized FAM classifiers. The top-performing classifiers from each population were then assembled into a classifier ensemble and compared against each other and against other classifier ensemble methods in the literature. Although the HFCPGA ensembles outperform GA ensembles in a majority of the tested cases, the slight increase in ensemble accuracy may not be justified given the significant processing time required for FAM optimization.

For further work, other selection methods with lower computation cost and better ensemble performance will be explored.

Acknowledgments This study was supported by University of Malaya Research Grant CG008-2013 and (UMRG) Project RG059-11ICT.

References

1. Carpenter GA, Grossberg S, Marzukon N, Reynolds JH, Rosen DB (1992) Fuzzy ARTMAP: a neural network architecture for incremental supervised learning of analog multidimensional maps. IEEE Trans Neural Netw 3(5):698–713
2. Jin M, Li R, Xu Z, Zhao X (2014) Reliable fault diagnosis method using ensemble Fuzzy ARTMAP based on improved Bayesian belief method. Neurocomputing 133(10):309–316
3. Mokhtar M, Howe J (2013) Comparing the online learning capabilities of Gaussian ARTMAP and Fuzzy ARTMAP for building energy management systems. Expert Syst Appl 40(15): 6007–6018
4. Tran VT, AlThobiani F, Ball A, Choi BK (2013) An application to transient current signal based induction motor fault diagnosis of Fourier–Bessel expansion and simplified fuzzy ARTMAP. Expert Syst Appl 40(13):5372–5384
5. Liang Z, Xu B, Chi Z, Feng D (2012) Intelligent characterization and evaluation of yarn surface appearance using saliency map analysis, wavelet transform and fuzzy ARTMAP neural network. Expert Syst Appl 39(4):4201–4212
6. Tan SC, Lim CP (2010) Evolutionary fuzzy artmap neural networks and their applications to fault detection and diagnosis. Neural Process Lett 31(3):219–242
7. Carpenter GA, Gaddam SC (2010) Biased ART: a neural architecture that shifts attention toward previously disregarded features following an incorrect prediction. Neural Netw 23(3): 435–451
8. Dagher I, Georgiopoulos M, Heileman GL, Bebis G (1999) An ordering algorithm for pattern presentation in fuzzy ARTMAP that tends to improve generalization performance. IEEE Trans Neural Netw 10(4):768–778
9. Al-Daraiseh A, Georgiopoulos M, Anagnostopoulos G, Wu AS, Mollaghasemi M (2006) GFAM: a genetic algorithm optimization of fuzzy ARTMAP. In: Proceedings of the 2006 IEEE International Conference on Fuzzy Systems, pp 315–322
10. Granger E, Henniges P, Oliveira L, Sabourin R (2006) Particle swarm optimization of fuzzy ARTMAP parameters. In: Proceedings of the 2006 International Joint Conference on Neural Networks, pp 2060–2067
11. Palaniappan R, Eswaran C (2009) Using genetic algorithm to select the presentation order of training patterns that improves simplified fuzzy ARTMAP classification performance. Appl Soft Comput 9(1):100–106
12. Cervantes L, Lee J, Lee J (2007) Agent-based approach to distributed ensemble learning of Fuzzy ARTMAP classifiers. In: Jo GS et al (eds) Agent and multi-agent systems: technologies and applications. Springer, Berlin, pp 805–814
13. de Medeiros Santos A, de Paula Canuto AM (2008) Using ARTMAP-based ensemble systems designed by three variants of boosting. In: Kurkova-Pohlova V et al (eds) Artificial neural networks—ICANN 2008. Springer, Berlin, pp 562–571
14. Williamson JR (1996) Gaussian ARTMAP: a neural network for fast incremental learning of noisy multidimensional maps. Neural Netw 9(5):881–897
15. Gómez-Sánchez E, Dimitriadis YA, Cano-Izquierdo JM, López-Coronado J (2002) μARTMAP: use of mutual information for category reduction in fuzzy ARTMAP. IEEE Trans Neural Netw 13(1):58–69
16. Kaylani A, Georgiopoulos M, Mollaghasemi M, Anagnostopoulos GC (2008) MO-GART: multiobjective genetic ART architectures. In: Proceedings of the 2008 IEEE Congress on Evolutionary Computation, pp 1425–1432
17. Zhang Y, Ji H, Zhang W (2014) TPPFAM: use of threshold and posterior probability for category reduction in fuzzy ARTMAP. Neurocomputing 124:63–71
18. Fernández C, Valle C, Saravia F, Allende H (2012) Behavior analysis of neural network ensemble algorithm on a virtual machine cluster. Neural Comput Appl 21(3):535–542

19. Seera M, Lim CP, Nahavandi S, Loo CK (2014) Condition monitoring of induction motors: a review and an application of an ensemble of hybrid intelligent models. Expert Syst Appl 41(10):4891–4903
20. Seera M, Lim CP (2014) A hybrid intelligent system for medical data classification. Expert Syst Appl 41(5):2239–2249
21. Sesmero MP, Alonso-Weber JM, Gutiérrez G, Ledezma A, Sanchis A (2012) A new artificial neural network ensemble based on feature selection and class recoding. Neural Comput Appl 21(4):771–783
22. Hu JJ, Goodman ED (2002) The hierarchical fair-competition (HFC) model for parallel evolutionary algorithms. In: Proceedings of the 2002 IEEE Congress on Evolutionary Computation, pp 49–54
23. Zenobi G, Cunningham P (2001) Using diversity in preparing ensembles of classifiers based on different feature subsets to minimize generalization error. In: de Raedt L et al. (eds) Machine learning: ECML 2001, pp 576–587
24. Yang J, Honavar V (1998) Feature subset selection using a genetic algorithm. In: Liu H et al. (eds) Feature extraction, construction and selection, pp 117–136
25. Opitz DW (1999) Feature selection for ensembles. In: Proceedings of the 16th National Conference on Artificial Intelligence (AAAI-99), pp 379–384
26. Cunningham P (2000) Overfitting and diversity in classification ensembles based on feature selection. In: Computer Science Technical Report TCD-CS-2000-07, Trinity College Dublin, pp 1–8
27. Oliveira LS, Morita M, Sabourin R (2006) Feature selection for ensembles using the multi-objective optimization approach. In: Jin Y (ed) Multi-objective machine learning. Springer, Berlin, pp 49–74
28. Canuto AMP, Vale KMO, Feitosa A (2011) A reinforcement-based mechanism to select features for classifiers in ensemble systems. Int J Comput Inf Syst Ind Manag Appl 3:324–335
29. Lin X, Yacoub S, Burns J, Simske S (2003) Performance analysis of pattern classifier combination by plurality voting. Pattern Recognit Lett 24(12):1959–1969
30. Loo CK, Rao M (2005) Accurate and reliable diagnosis and classification using probabilistic ensemble simplified fuzzy ARTMAP. IEEE Trans Knowl Data Eng 17(11):1589–1593
31. Radtke PV, Sabourin R, Wong T, et al. (2006) Classification system optimization with multi-objective genetic algorithms. In: Tenth International Workshop on Frontiers in Handwriting Recognition
32. Ishibuchi H, Yamamoto T (2003) Evolutionary multiobjective optimization for generating an ensemble of fuzzy rule-based classifiers. In: Cantu-Paz E et al (eds) Genetic and evolutionary computation GECCO 2003. Springer, Berlin, pp 1077–1088
33. Mohamed MA, Hegazy AEF, Badr AA (2011) Evolutionary fuzzy ARTMAP approach for breast cancer diagnosis. Int J Comput Sci Netw Secur 11(4):77–84
34. Qasem SN, Shamsuddin SM, Zain AM (2012) Multi-objective hybrid evolutionary algorithms for radial basis function neural network design. Knowledge-Based Syst 27:475–497
35. Wu CH, Tzeng GH, Lin RH (2009) A Novel hybrid genetic algorithm for kernel function and parameter optimization in support vector regression. Expert Syst Appl 36:4725–4735
36. Frank A, Asuncion A (2011) UCI machine learning repository. http://archive.ics.uci.edu/ml
37. Dzeroski S, Zenko B (2002) Is combining classifiers better than selecting the best one? In: International Workshop and Conference on Machine Learning, pp 123–130
38. Partalas I, Tsoumakas G, Katakis I, Vlahavas I (2006) Ensemble pruning using reinforcement learning. In: Advances in artificial intelligence, Springer, pp 301–310
39. Zhang Y, Burer S, Street WN (2006) Ensemble pruning via semi-definite programming. J Mach Learn Res 7:1315–1338
40. Ordonez FJ, Ledezma A, Sanchis A (2008) Genetic approach for optimizing ensembles of classifiers. In: Proceedings of the Twenty-First International Florida Artificial Intelligence Research Society Conference (FLAIRS) Conference, pp 89–94
41. Wang SJ, Mathew A, Chen Y, Xi LF, Ma L, Lee J (2009) Empirical analysis of support vector machine ensemble classifiers. Expert Syst Appl 36(3):6466–6476
42. Wilcoxon F (1945) Individual comparisons by ranking methods. Biometrics 1(6):80–83
43. Dos Santos EM, Sabourin R, Maupin P (2009) Overfitting cautious selection of classifier ensembles with genetic algorithms. Inf Fus 10(2):150–162
44. Lofstrom T, Johansson U, Bostrom H (2009) Ensemble member selection using multi-objective optimization. In: IEEE 2009 Symposium on Computational Intelligence and Data Mining, IEEE, pp 245–251
45. Connolly JF, Granger E, Sabourin R (2012) Evolution of heterogeneous ensembles through dynamic particle swarm optimization for video-based face recognition. Pattern Recognit 45(7):2460–2477
46. Liu Y, Yao X, Higuchi T (2000) Evolutionary ensembles with negative correlation learning. IEEE Trans Evol Comput 4(4):380–387